CLAIMS

1. A system for controlling co-scheduling of processes in a computer comprising at least one process and a spin daemon, the process being configured to, when it is waiting for a flag to change condition, transmit a flag monitor request to the spin daemon and de-schedule itself, the spin daemon being configured to, after receiving a flag monitor request monitor the flag and, after the flag changes condition, enable the at least one process to be re-scheduled for execution by the computer.

2. A system as defined in claim 1 in which said spin daemon is configured to monitor a plurality of flags, each in response to a flag monitor request, the spin daemon maintaining a list identifying those flags it is to monitor, the spin daemon being further configured to, when it receives a flag monitor request, add an identification of a flag associated with the request to the list.

3. A system as defined in claim 2 in which said flags are contained in a memory segment, the spin daemon being configured to enable the at least one process to be re-scheduled following a change of condition of any flag in said memory segment.

4. A system as defined in claim 1 in which said at least one process is configured to register with said spin daemon, during registration the at least one process being configured to provide the spin daemon with an identifier for the memory segment, the spin daemon being configured to provide a handle, the at least one process being configured to use the handle in the flag monitor request.

5. A system as defined in claim 1 in which said at least one process and said spin daemon are configured to communicate over a socket.

6. A method of controlling co-scheduling of processes in a computer comprising at least one process and a spin daemon, the method comprising the steps of:

A. enabling the process to, when it is waiting for a flag to change condition, transmit a flag monitor request to the spin daemon and de-schedule itself,

B. enabling the spin daemon to, after receiving a flag monitor request monitor the flag and, after the flag changes condition, enable the at least one process to be re-scheduled for execution by the computer.

7. A method as defined in claim 6, the spin daemon being configured to monitor a plurality of flags, each in response to a flag monitor request, the spin daemon maintaining a list identifying those flags it is to monitor, the method including the step of enabling the spin daemon being to, when it receives a flag monitor request, add an identification of a flag associated with the request to the list.

8. A method as defined in claim 7 in which said flags are contained in a memory segment, the method including the step of enabling the spin daemon to enable the at least one process to be re-scheduled following a change of condition of any flag in said memory segment.

9. A method as defined in claim 6 further including the steps of

A enabling the at least one process to register with said spin daemon, during registration the at least one process being configured to provide the spin daemon with an identifier for the memory segment; and

B. enabling the spin daemon to provide a handle for use by the at least one process in the flag monitor request.

10. A method as defined in claim 6 further comprising the step of enabling the at least one process and said spin daemon communicate over a socket.

11. A computer program product for use in connection with a computer to control co-scheduling of at least one process in the computer, the computer program product including a computer readable medium having encoded thereon:

A. a process module configured to enable the computer to, when the process is waiting for a flag to change condition, transmit a flag monitor request to the spin daemon and de-schedule itself,

B. a spin daemon module configured to enable the computer to, after receiving a flag monitor request monitor the flag and, after the flag changes condition, enable the at least one process to be re-scheduled for execution by the computer.

12. A computer program product as defined in claim 11 in which said spin daemon is configured to enable the computer to monitor a plurality of flags, each in response to a flag monitor request, the spin daemon enabling the computer to maintain a list identifying those flags it is to monitor, the spin

daemon being further configured to enable the computer to, when it receives a flag monitor request, add an identification of a flag associated with the request to the list.

13. A computer program product as defined in claim 12 in which said flags are contained in a memory segment, the spin daemon being configured to enable the computer enable the at least one process to be re-scheduled following a change of condition of any flag in said memory segment.

14. A cp as defined in claim 11 in which said at least one process is configured to enable the computer to register with said spin daemon, during registration the at least one process being configured to enable the computer to provide the spin daemon with an identifier for the memory segment, the spin daemon being configured to enable the computer to provide a handle, the at least one process being configured to use the handle in the flag monitor request.

15. A computer program product as defined in claim 11 in which said at least one process and said spin daemon are configured to enable the computer to communicate over a socket.